Project 4 Programmer Manual

1. Problem Description

Menu/Prompt-driven test driver for Huffman algorithm.
This program uses the Huffman coding algorithm for file compression and decompression.

2. Data Types and Classes:

The data types used in this program fall into two categories: predefined data types and programmer-defined datatypes. The following subsections address the data types used.

2.1 Programmer-defined Type:

- **info:**

variable **info:**

- A node inside a Huffman encoding tree. Each node stores four values:
- the character,
- pointers to the 0 and 1 subtrees,
- and the character count(weight) of the tree.

The object has a constructor**:** construct a new info node to store the given letter and its count, along with the given child pointers.

2.2 Data Types:

- **`info*:`**

variable huffTree: Huffman tree
variable encodingTree: Huffman tree.

- **`list<info*>:`**

variable freqTable: frequency table.

- **`ifstream:`**

variable inFile: the stream to be connected to the input file.

- **`ofstream:`**

variable outFile:	the stream to be connected to the output file.

- **`string:`**

variable choice:	menu option
variable textStr:	text data store in file.
variable huffTablePath:	Huffman table file path.
variable encodePath:	file path to store encoded msg
fileInput:	input file path from user input.

---

3. High Level Program Solution

3.1 Algorithm:

Main program:

- Call intro function for printing program introduction
- Print menu options
- If user enters 'Q' or 'q',
  - exit the program.
- If user enter 1, then encode data:
  - generate Huffman table from file
  - generate compressed file
- If user enter 2, then decode data:
  - generate Huffman table from encode
  - generate decompress file

3.2 Function Headings:

- Function: *void* verifyInput(T& input)

// Input option validation

- Function: void intro()

// program purpose

- *Function:* string menu()

// Main Menu

- Function: string readFile(ifstream& inFile, string prompt)

// read file from user input

- Function: string writeFile(ofstream& outFile, string prompt)

// output file to specified path

- *Function: void* printFreqTable(*const* list<info*>& freqTable)

// print frequency table

- *Function: void* test_huffmanTable(string& textStr, string& huffTablePath)

// test frequency table and Huffman table

- *Function: void* test_encodeData(*const* string& textStr, *const* string& huffTablePath)

// test compression

- *Function: void* text_decodeData()

// test decompression